Figure 1

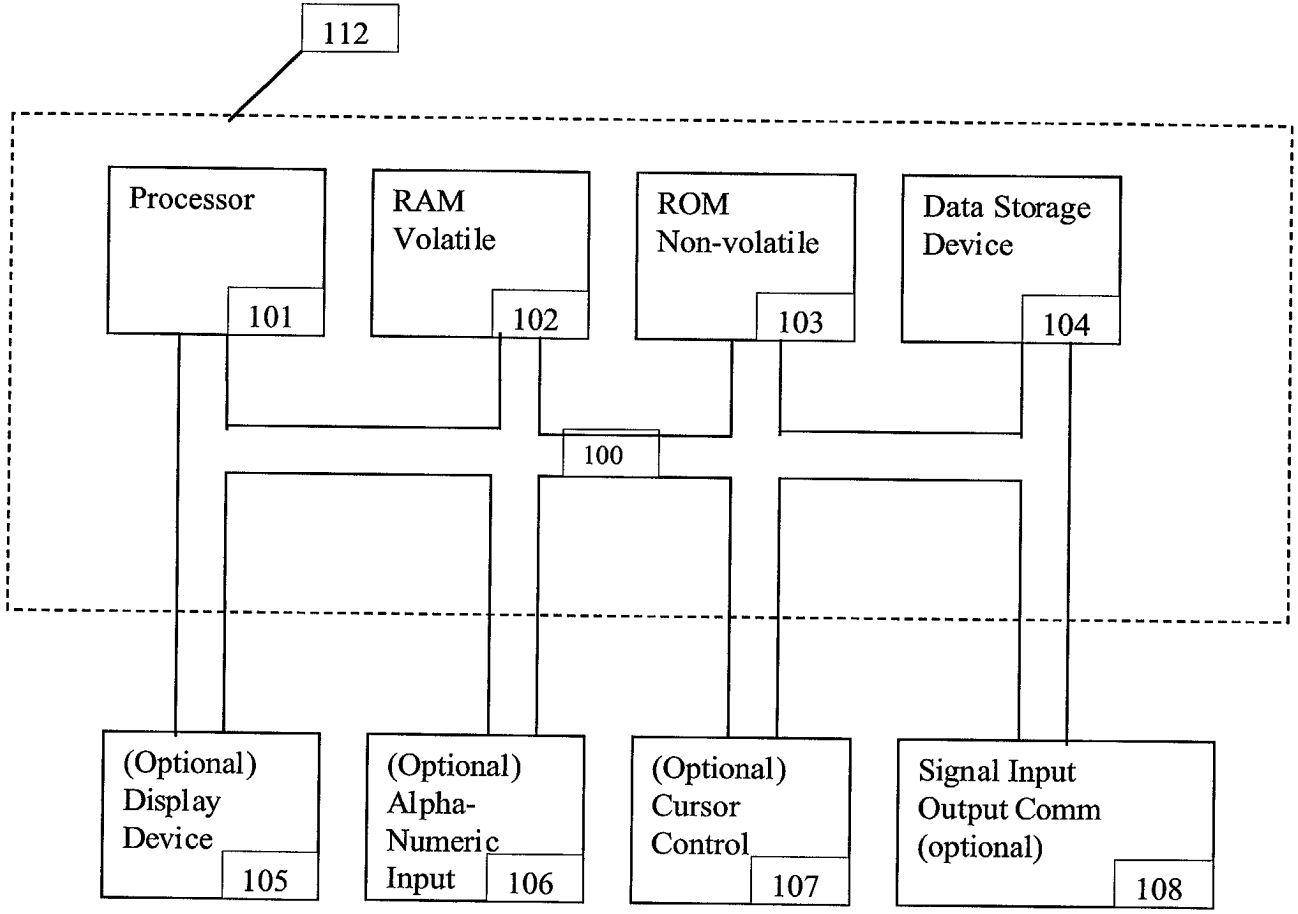
112
Processor
101
RAM
Volatile
102
ROM
Non-volatile
103
Data Storage
Device
104
100
(Optional)
Display
Device
105
(Optional)
Alpha-
Numeric
Input
106
(Optional)
Cursor
Control
107
Signal Input
Output Comm
(optional)
108

Figure 2

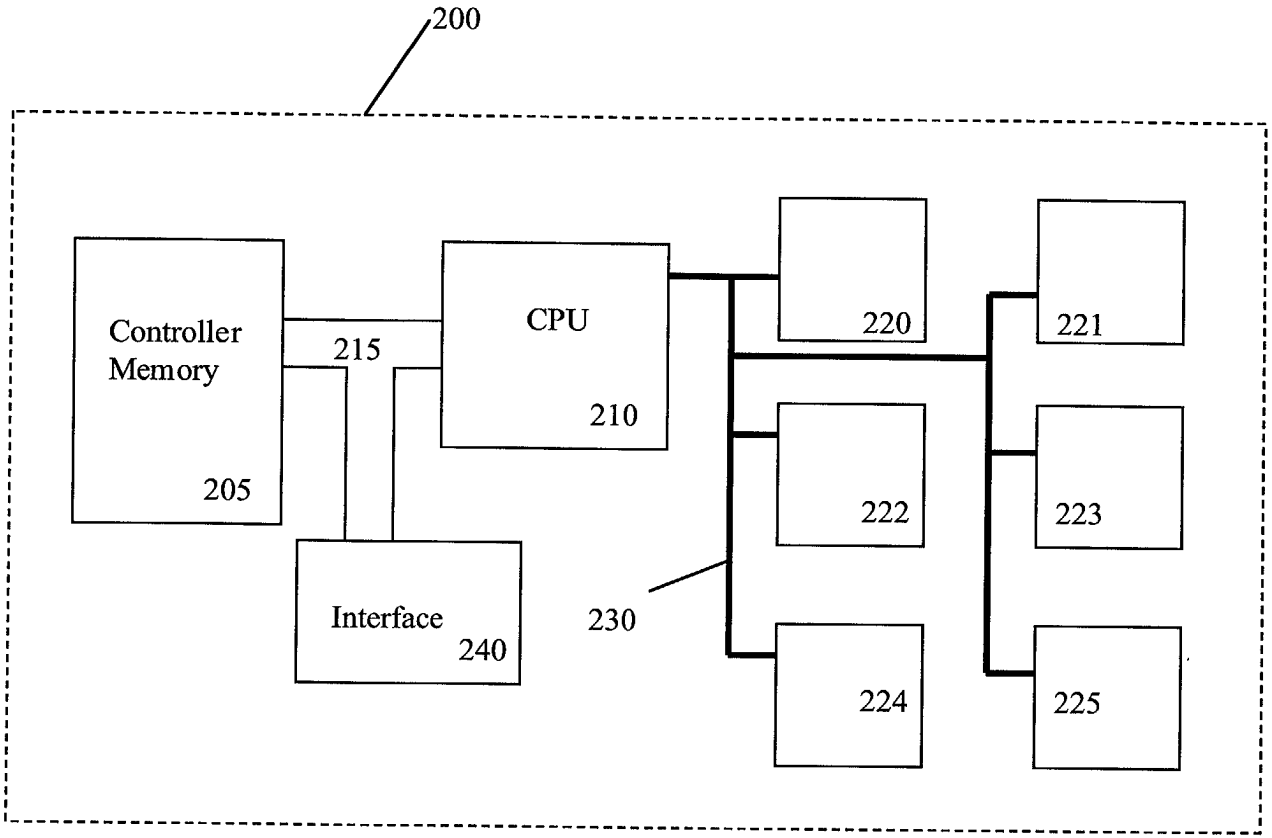
200
Controller Memory
205
215
CPU
210
220
221
222
223
224
225
230
Interface
240

Figure 3

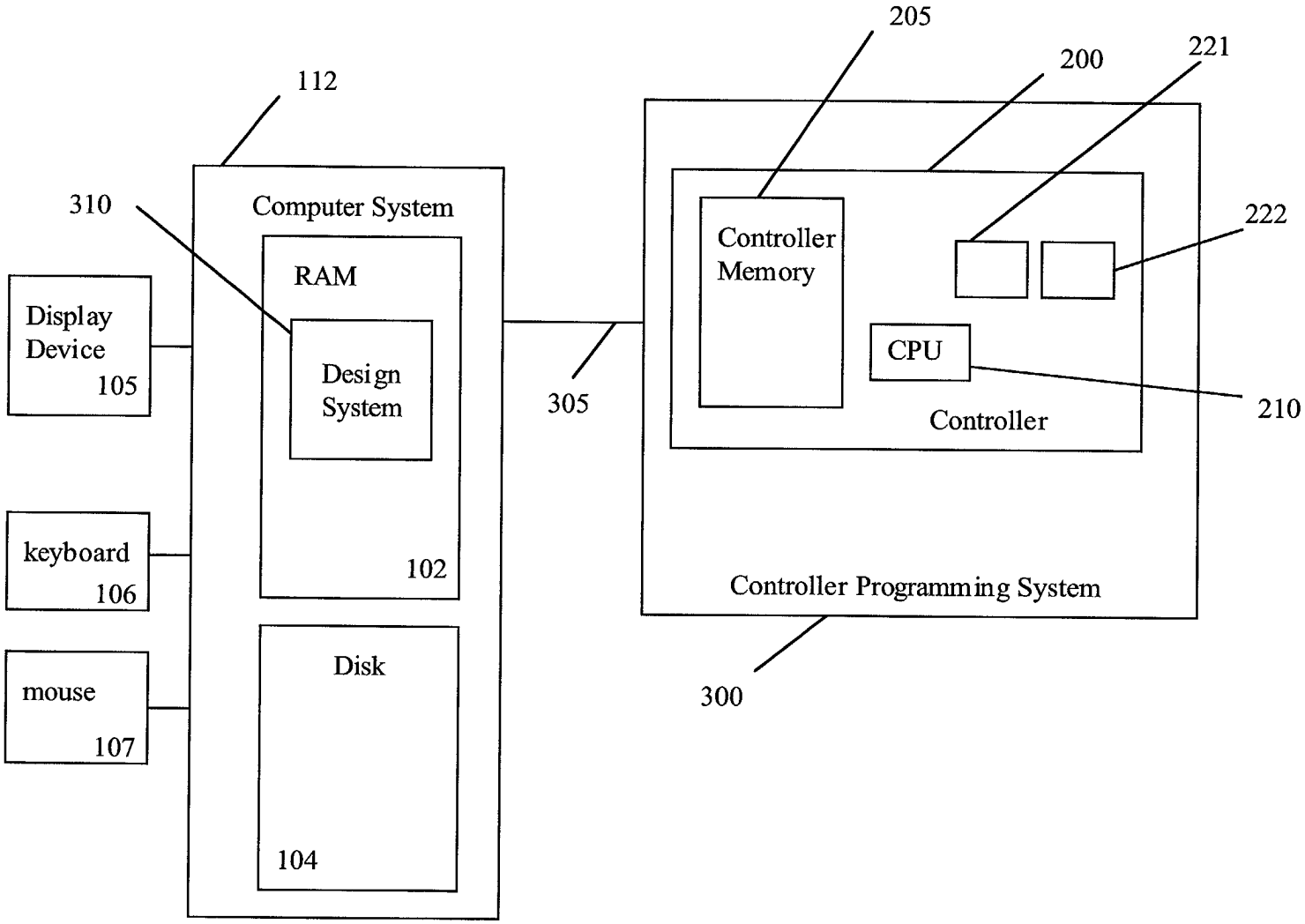
205
200
221
112
310
Computer System
RAM
Design System
102
Display Device
105
keyboard
106
mouse
107
Disk
104
305
Controller Memory
CPU
222
210
Controller
Controller Programming System
300

Figure 4

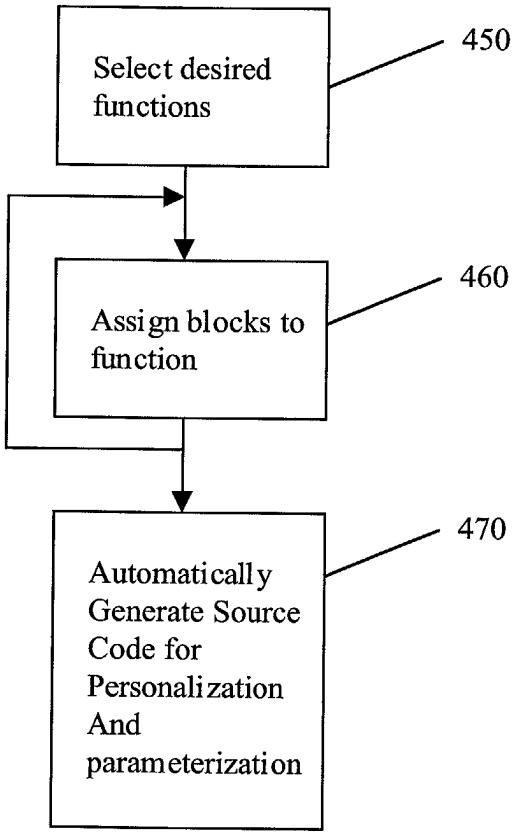
Select desired functions
450
Assign blocks to function
460
Automatically Generate Source Code for Personalization And parameterization
470

Figure 5

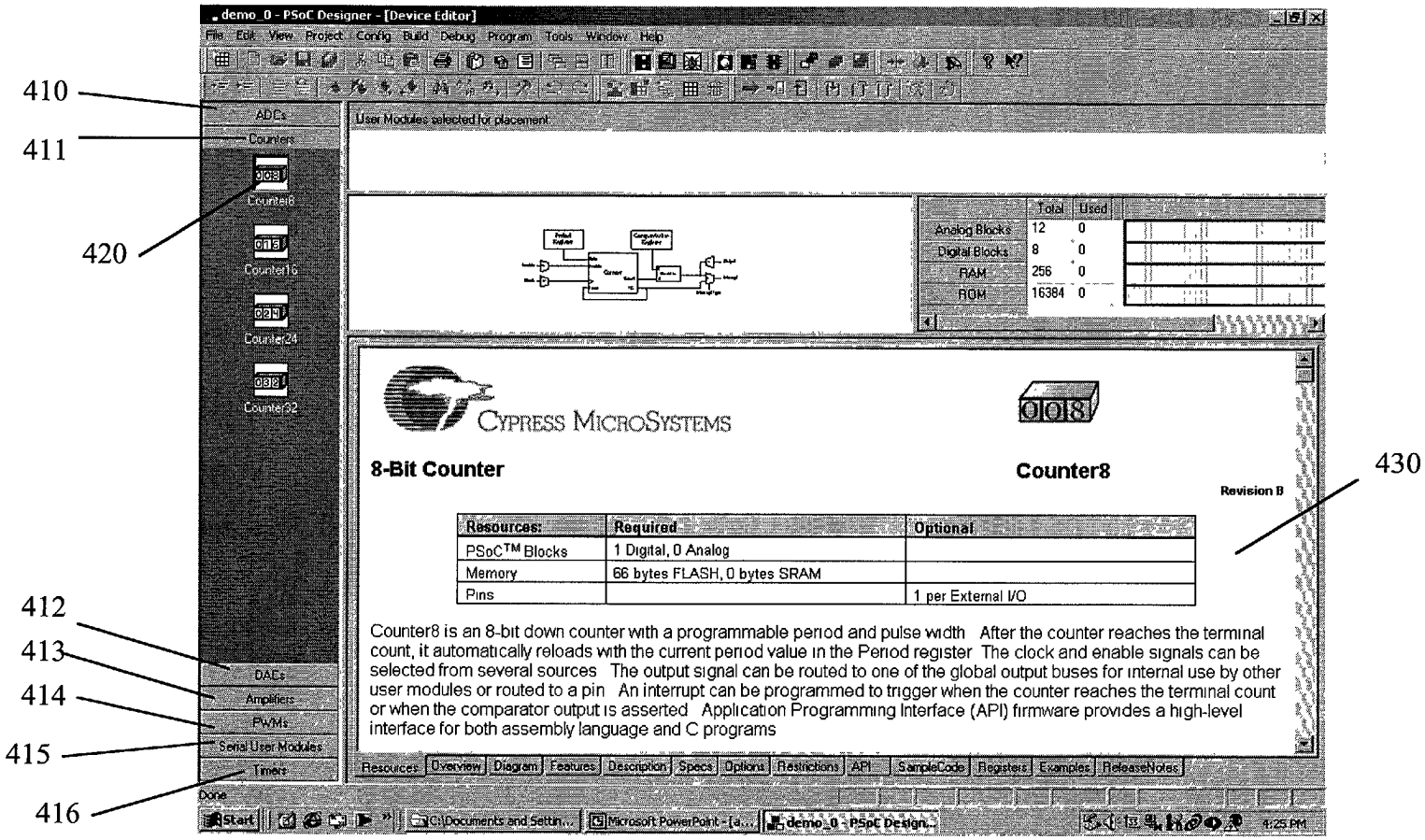
410
411
420
412
413
414
415
416
430
demo_0 - PSoC Designer - [Device Editor]
File Edit View Project Config Build Debug Program Tools Window Help
ADCs
Counters
Counter8
Counter16
Counter24
Counter32
DACs
Amplifiers
PWMs
Serial User Modules
Timers
User Modules selected for placement
Total Used
Analog Blocks 12 0
Digital Blocks 8 0
RAM 256 0
ROM 16384 0
CYPRESS MICROSYSTEMS
8-Bit Counter
Counter8
Revision B
Resources: Required Optional
PSoC™ Blocks 1 Digital, 0 Analog
Memory 66 bytes FLASH, 0 bytes SRAM
Pins 1 per External I/O
Counter8 is an 8-bit down counter with a programmable period and pulse width After the counter reaches the terminal count, it automatically reloads with the current period value in the Period register The clock and enable signals can be selected from several sources The output signal can be routed to one of the global output buses for internal use by other user modules or routed to a pin An interrupt can be programmed to trigger when the counter reaches the terminal count or when the comparator output is asserted Application Programming Interface (API) firmware provides a high-level interface for both assembly language and C programs
Resources Overview Diagram Features Description Specs Options Restrictions API SampleCode Registers Examples ReleaseNotes
Done
Start
C:\Documents and Settin...
Microsoft PowerPoint - [a...
demo_0 - PSoC Design...
4:25 PM

Figure 6

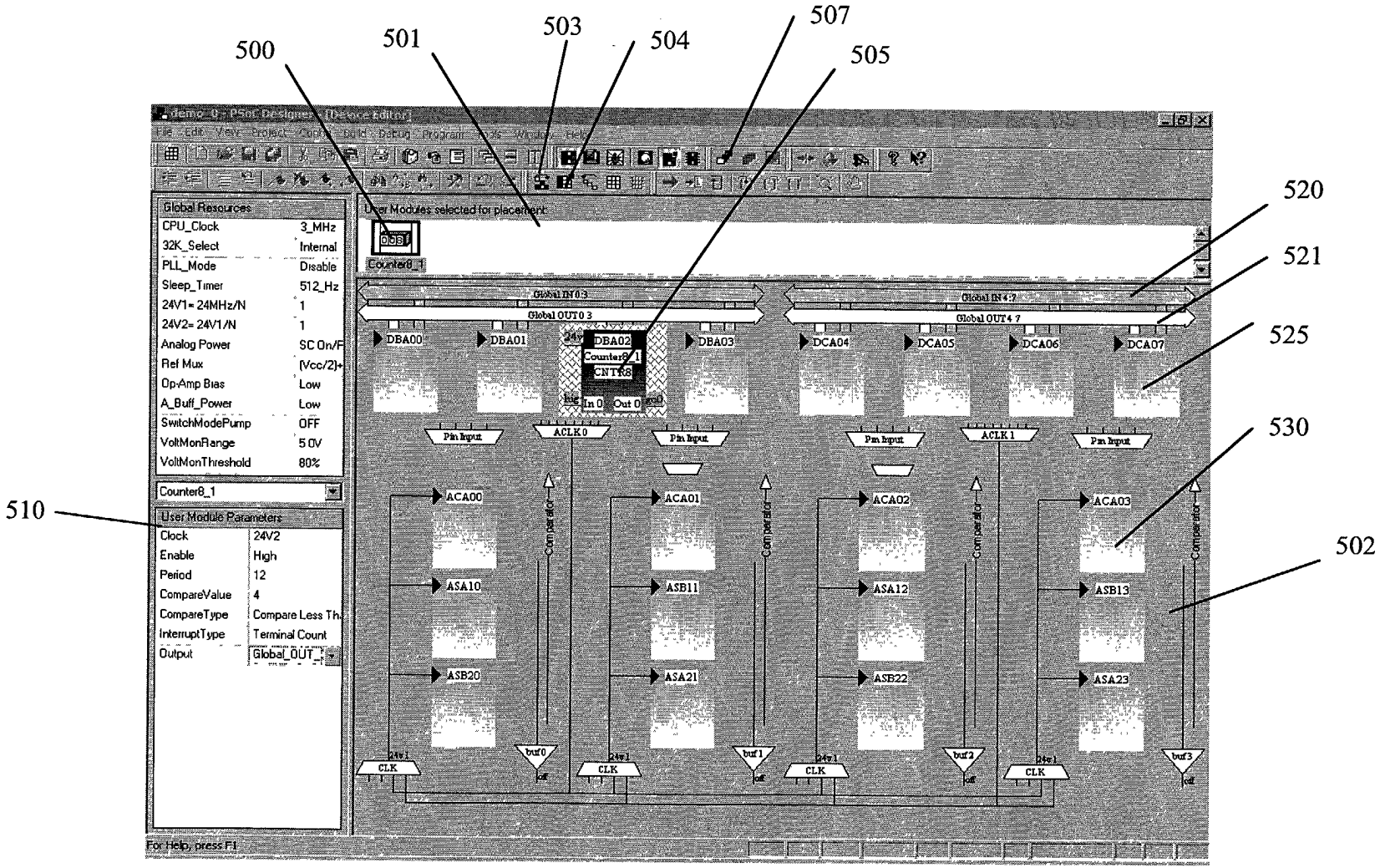
500
501
503
504
507
505
520
521
525
530
502
510
Global Resources
CPU_Clock 3_MHz
32K_Select Internal
PLL_Mode Disable
Sleep_Timer 512_Hz
24V1= 24MHz/N 1
24V2= 24V1/N 1
Analog Power SC On/F
Ref Mux (Vcc/2)+
Op-Amp Bias Low
A_Buff_Power Low
SwitchModePump OFF
VoltMonRange 5.0V
VoltMonThreshold 80%
Counter8_1
User Module Parameters
Clock 24V2
Enable High
Period 12
CompareValue 4
CompareType Compare Less Th
InterruptType Terminal Count
Output Global_OUT_
User Modules selected for placement
Counter8_1
Global IN 0 3
Global OUT 0 3
Global IN 4 7
Global OUT 4 7
DBA00
DBA01
DBA02
DBA03
DCA04
DCA05
DCA06
DCA07
Counter8_1
CNT<8
In 0
Out 0
Pin Input
ACLK 0
ACLK 1
ACA00
ACA01
ACA02
ACA03
ASA10
ASB11
ASA12
ASB13
ASB20
ASA21
ASB22
ASA23
Comparator
buf0
buf1
buf2
buf3
24v1
CLK
For Help, press F1

Figure 7

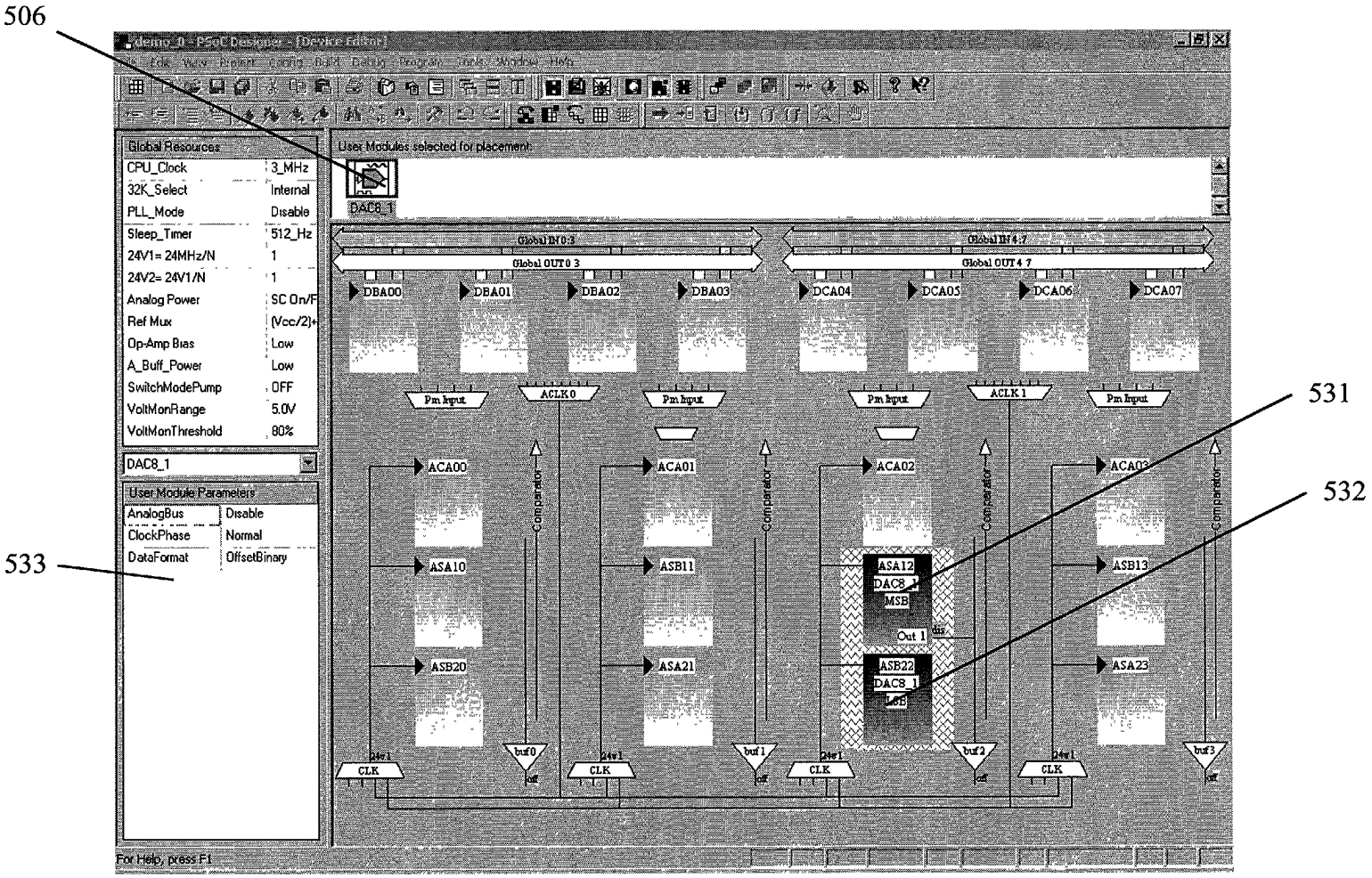
506
531
532
533
Global Resources
CPU_Clock 3_MHz
32K_Select Internal
PLL_Mode Disable
Sleep_Timer 512_Hz
24V1= 24MHz/N 1
24V2= 24V1/N 1
Analog Power SC On/F
Ref Mux (Vcc/2)+
Op-Amp Bias Low
A_Buff_Power Low
SwitchModePump OFF
VoltMonRange 5.0V
VoltMonThreshold 80%
DAC8_1
User Module Parameters
AnalogBus Disable
ClockPhase Normal
DataFormat OffsetBinary
User Modules selected for placement:
DAC8_1
Global OUT 0 3
Global OUT 4 7
DBA00
DBA01
DBA02
DBA03
DCA04
DCA05
DCA06
DCA07
Pm Input
ACLK 0
ACLK 1
ACA00
ACA01
ACA02
ACA03
ASA10
ASB11
ASA12
DAC8_1
MSB
Out 1
ASB13
ASB20
ASA21
ASB22
DAC8_1
LSB
ASA23
Comparator
buf0
buf1
buf2
buf3
24v1
CLK
For Help, press F1

Figure 8A

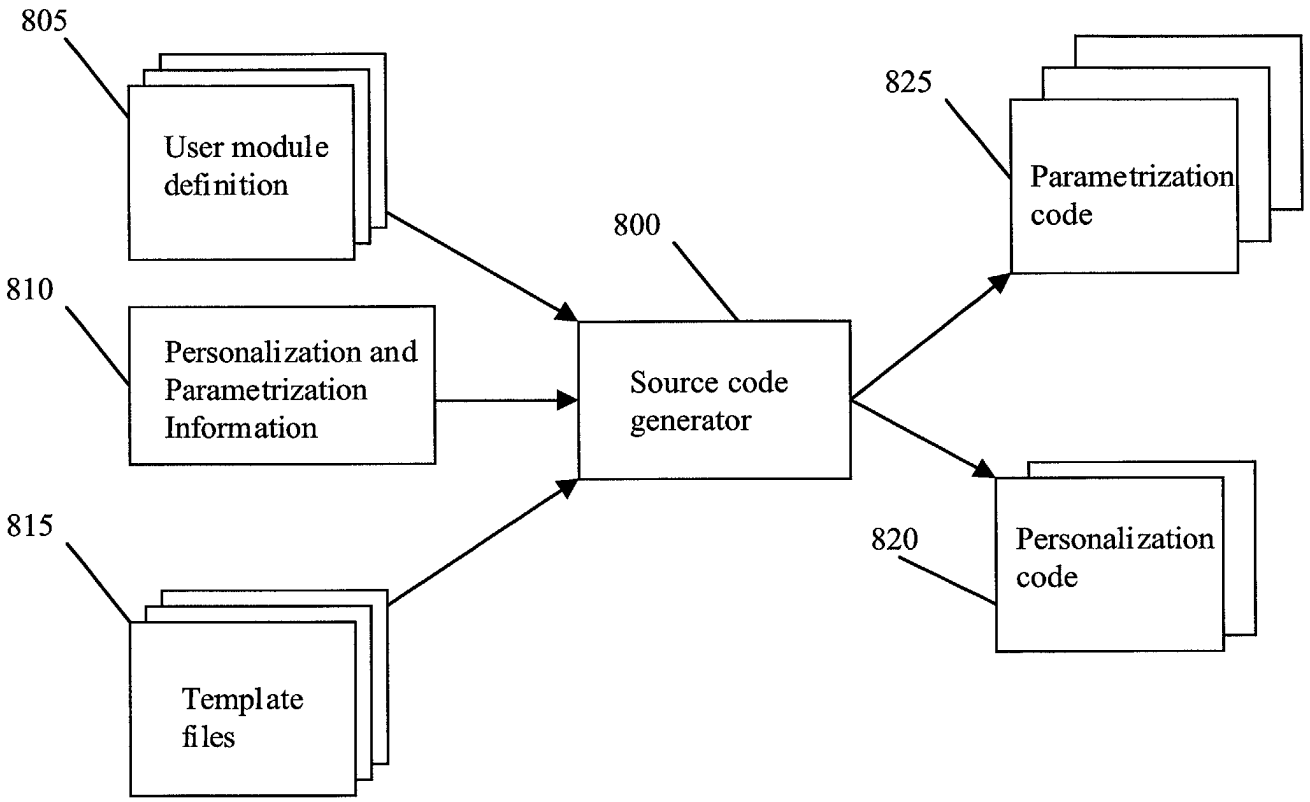
805
User module definition
810
Personalization and Parametrization Information
815
Template files
800
Source code generator
825
Parametrization code
820
Personalization code

Figure 8B

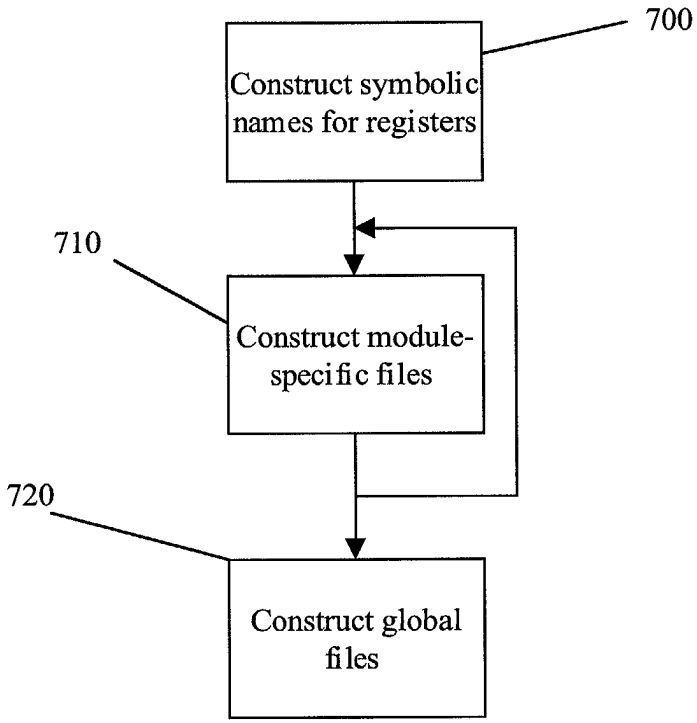
700
Construct symbolic names for registers
710
Construct module-specific files
720
Construct global files

Figure 9

```
;-----------------------------------
; Interrupt Vector Table
;-----------------------------------
; Interrupt vector table entries are 4 bytes long and contain the code
; that services the interrupt (or causes it to be serviced).
;-----------------------------------
  AREA   TOP(ROM, ABS)

  org 0                ; Reset Interrupt Vector
  jmp __start          ; First instruction executed following a Reset

  org 04h              ; Supply Monitor Interrupt Vector
  `@INTERRUPT_1`
  reti

  org 08h              ; PSoC Block DBA00 Interrupt Vector
  `@INTERRUPT_2`
  reti
```

633

Figure 10

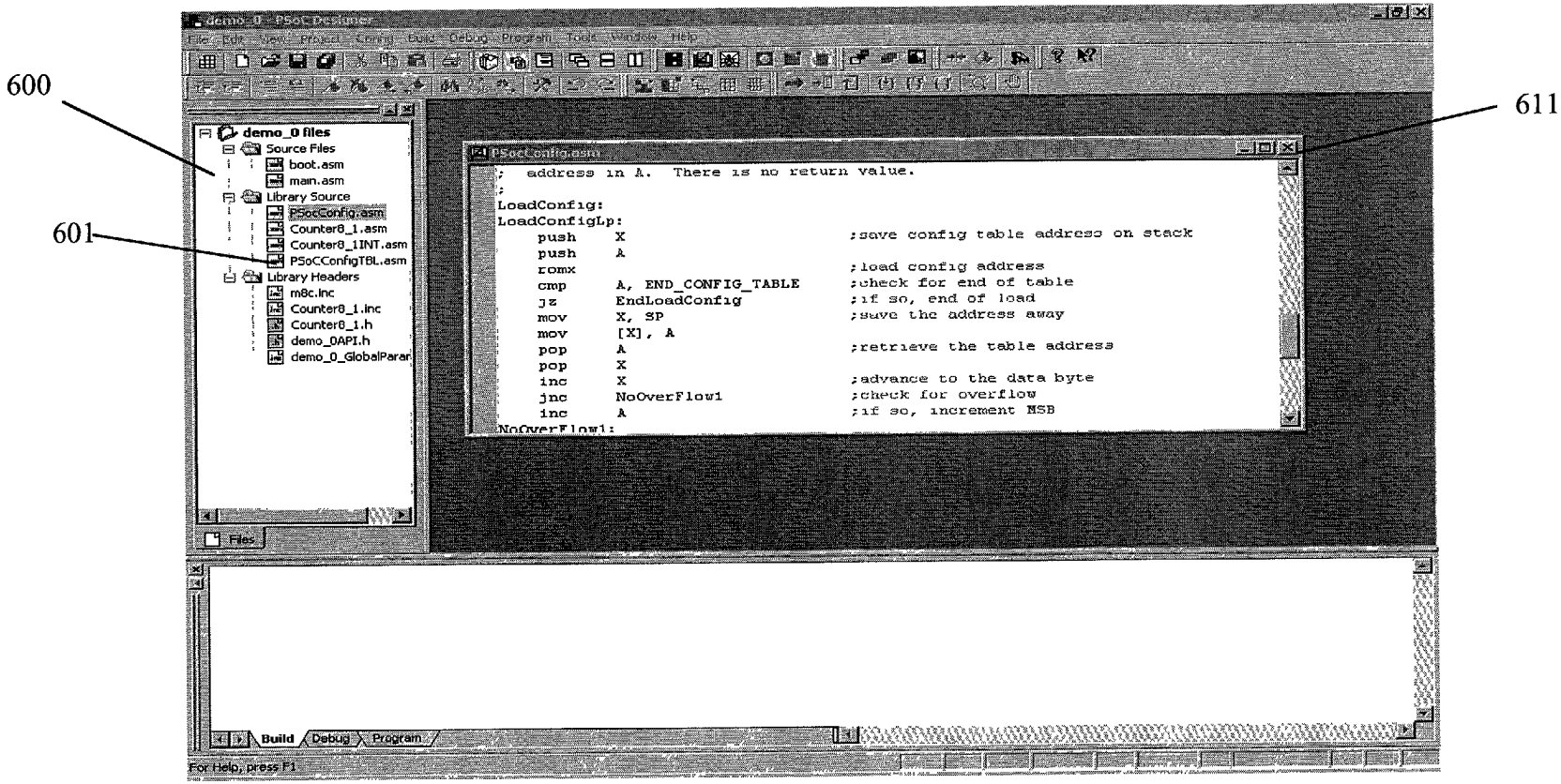
600
601
611
demo_0 files
Source Files
boot.asm
main.asm
Library Source
PSoCConfig.asm
Counter8_1.asm
Counter8_1INT.asm
PSoCConfigTBL.asm
Library Headers
m8c.inc
Counter8_1.inc
Counter8_1.h
demo_0API.h
demo_0_GlobalParar
Files
; address in A. There is no return value.
;
LoadConfig:
LoadConfigLp:
push X ;save config table address on stack
push A
romx ;load config address
cmp A, END_CONFIG_TABLE ;check for end of table
jz EndLoadConfig ;if so, end of load
mov X, SP ;save the address away
mov [X], A
pop A ;retrieve the table address
pop X
inc X ;advance to the data byte
jnc NoOverFlow1 ;check for overflow
inc A ;if so, increment MSB
NoOverFlow1:
Build
Debug
Program
For Help, press F1

Figure 11

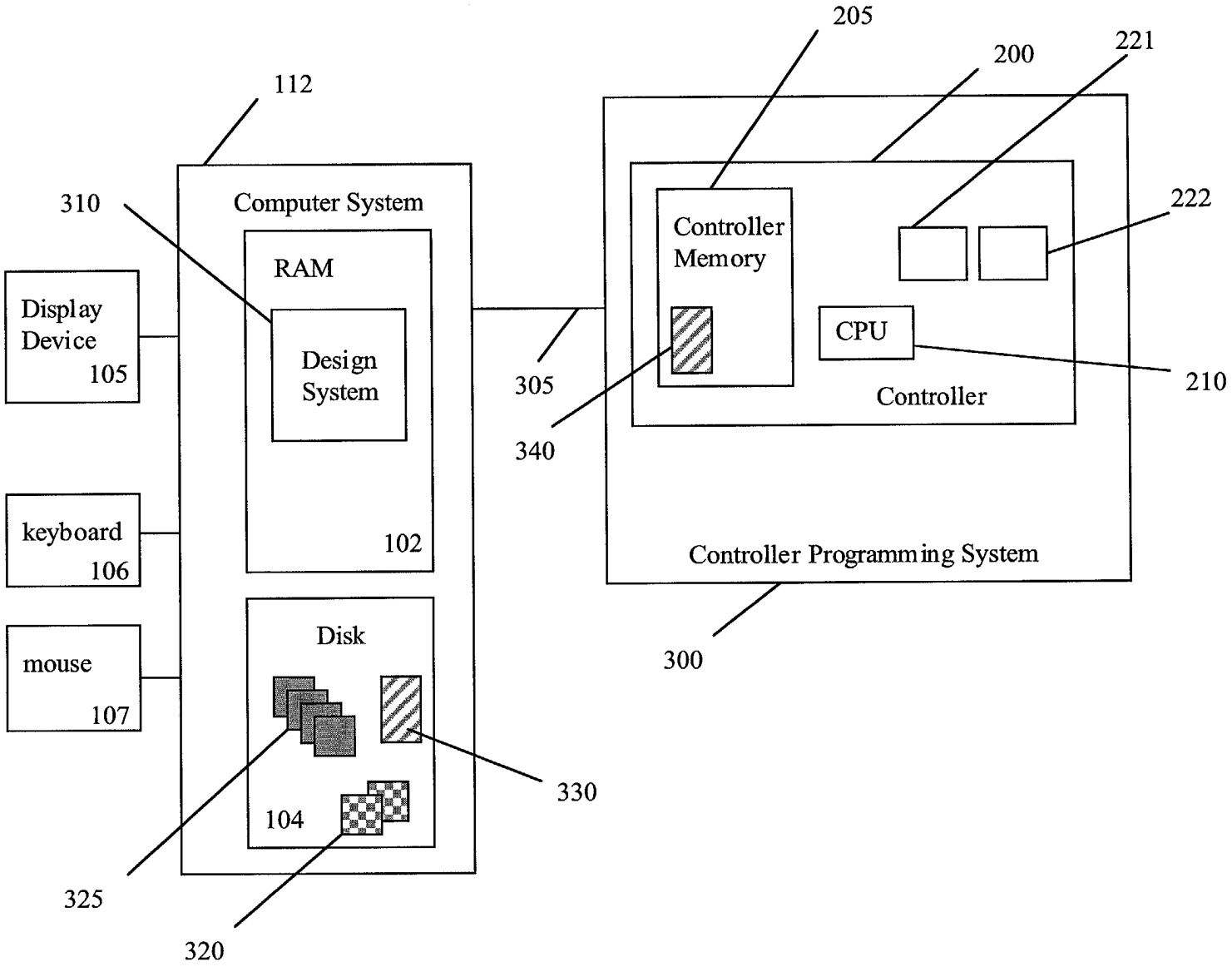
205
200
221
112
310
Computer System
RAM
Design System
102
Display Device 105
keyboard 106
mouse 107
Disk
104
325
320
330
305
340
Controller Memory
CPU
222
210
Controller
Controller Programming System
300

Figure 12

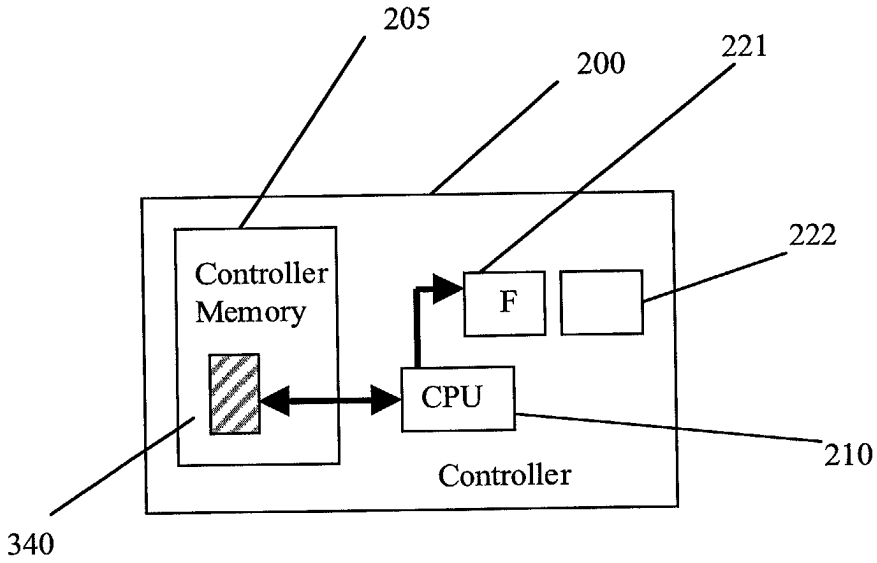

Figure 13

```
LoadConfigTBL_demo_0_Bank0:
; Global Register values
        db      60h, 28h        ; AnalogColumnInputSelect register
        db      63h, 05h        ; AnalogReferenceControl register
        db      65h, 00h        ; AnalogSyncControl register
        db      e6h, 00h        ; DecimatorControl register
        db      02h, 03h        ; Port_0_Bypass register
        db      06h, 00h        ; Port_1_Bypass register
        db      0ah, 00h        ; Port_2_Bypass register
        db      0eh, 00h        ; Port_3_Bypass register
        db      12h, 00h        ; Port_4_Bypass register
        db      16h, 00h        ; Port_5_Bypass register
; Instance name Counter8_1, User Module Counter8
;       Instance name Counter8_1, Block Name CNTR8(DBA02)
        db      2bh, 00h        ;Counter8_1_CONTROL_REG
        db      29h, 0ch        ;Counter8_1_PERIOD_REG
        db      2ah, 04h        ;Counter8_1_COMPARE_REG
        db      ffh
```

632

Figure 14

```
;
; THEORY of OPERATION:
;    Write data into the Period register.
;
;-----------------------------------------------------------------------
 Counter8_1_WritePeriod:
_Counter8_1_WritePeriod:
  mov  REG[Counter8_1_PERIOD_REG ], A
  ret
```

Figure 15

```
;--------------------------------
;  Registers used by counter8
;--------------------------------
Counter8_1_CONTROL_REG:     equ     2bh     ;Control register
Counter8_1_COUNTER_REG:     equ     28h     ;Counter register
Counter8_1_PERIOD_REG :     equ     29h     ;Period value register
Counter8_1_COMPARE_REG:     equ     2ah     ;CompareValue register
Counter8_1_FUNC_REG   :     equ     28h     ;Function register
Counter8_1_INPUT_REG  :     equ     29h     ;Input register
Counter8_1_OUTPUT_REG :     equ     2ah     ;Output register

; end of file
```

Figure 16

```
/***************************************************
* Prototypes of Counter8 API. For a definition of
* functions see Counter8_1.inc.
***************************************************/
extern void  Counter8_1_EnableInt(void);
extern void  Counter8_1_DisableInt(void);
extern void  Counter8_1_Start(void);
extern void  Counter8_1_Stop(void);
extern void  Counter8_1_WritePeriod(BYTE bPeriod);
extern void  Counter8_1_WriteCompareValue(BYTE bCompareValue);
extern BYTE bCounter8_1_ReadCompareValue(void);
extern BYTE bCounter8_1_ReadCounter(void);
```